

Cloud computing with Google Apps for education: An experience report

**Eder John Scheid, Leandro Hundertmarck Minato,
Benhur de Oliveira Stein, Andrea Schwertner Charão**

Universidade Federal de Santa Maria. Av. Roraima, 1000, 97105-900, Santa Maria, RS, Brazil
eder@inf.ufsm.br, minato@inf.ufsm.br, benhur@inf.ufsm.br, andrea@inf.ufsm.br

Abstract. This article presents an experience report on using Google Apps for Education in a computer science laboratory at the Federal University of Santa Maria, Brazil. Google Apps platform offers a range of applications in a SaaS (Software as a Service) cloud, bringing several facilities for members of the institution, but also some challenges for system administrators. Throughout the article, we describe the migration to the cloud platform, the current state of the migrated domain and some opportunities that are being explored to best meet our requirements.

Key words: cloud computing, Software as a Service, Google Apps, system administration.

Introduction

There is currently an increasing interest on services based on the cloud computing paradigm, where systems and applications are available anywhere in the world. Many companies now offer cloud services and Google is an example of them. Its SaaS (Software as a Service) platform, known as Google Apps, offers a large set of cloud-based applications for individuals, business, education and government. One of its branches is Google Apps for Education, a Google Apps solution offered to educational institutions without charge and without any ads.

At the end of 2007, one of the computer science laboratories of the Federal University of Santa Maria (*Universidade Federal de Santa Maria - UFSM*) began to adopt Google Apps for Education. This laboratory, known as NCC (*Núcleo de Ciência da Computação*) is responsible for the network domain `inf.ufsm.br` and offers various services to students, faculty and staff related to some computer science undergraduate and postgraduate courses at UFSM. The main motivation, at the time of migration to Google Apps, was to replace the email service that was managed locally and demanded a lot of effort from the NCC team. Since then, the management of cloud services is being carried

out by a team of students and professors using the interfaces offered by Google Apps for Education. Using this SaaS platform brought benefits to users and administrators. For the latter, there were also some new challenges and opportunities.

The main objective of this article is to report the experience gained by the staff of NCC with Google Apps for Education, providing some useful information to other institutions that plan to follow a similar path. The rest of the paper is organized as follows. The next section presents Google Apps for Education, its resources and objectives. Then, we present NCC and provide information about the services it offers to users in the `inf.ufsm.br` network domain. In the fourth section we describe the migration to Google Apps for Education process and our experience on managing our domain in recent years. It includes some observations on problems we face nowadays. After that, we discuss opportunities that are being explored as to automate some management tasks and, in the last section, we present some final remarks.

Google Apps for Education

The Google Apps platform (Google Inc., 2012c) offers several cloud applications, including email (Gmail), instant messaging

(Google Talk), calendar (Google Calendar), storage (Google Drive) and web sites (Google Sites). This platform has a branch focused on companies (Google Apps for Business), currently priced at US\$ 5/user/month (Google Inc., 2012a). This version allows the user base of a given domain to be managed via Google Apps, with Google servers responding to the services offered in this domain. Google Apps for Education (Google Inc., 2012b) is an edition of Google Apps platform with the same features of Google Apps for Business, but freely available to educational institutions.

Many institutions at various levels of education have benefited from cloud services with Google Apps for Education (Google Inc., 2012d). Furthermore, many studies have discussed Google Apps platform for educational institutions (Barlow and Lane, 2007; Herrick, 2009; Sultan, 2010). The free edition helps organizations that lack the resources to install and maintain such services locally, but also represents a successful strategy to IT teams so they can focus on other services targeting more significant activities in educational institutions.

Google Apps for Education has broad support to meet the concerns of users and system administrators (Google Inc., 2012f). For the latter, the platform provides detailed instructions for signing up to the service and for managing users and applications. Among the facilities for system administrators, there are administrative APIs that assist in the migration of the user database via synchronization with LDAP or Active Directory (Google Inc., 2012g).

In addition to all these resources, the platform also provides a simplified control panel

for system administrators (see Figure 1). In its home screen, the control panel provides a dashboard presenting information about the users that are registered in the domain (last login, an option to reset password and a button to create a new user). It also lists all Google Apps services that are enabled within the domain, like Gmail, Calendar, Drive and Docs, Gtalk and Contacts. Besides the dashboard, the control panel provides a series of utilities for system administrators such as management of users, organizations and groups, domain settings and other setup utilities, reports and a support section containing some documentation and answers to different questions about Google Apps.

One important feature present in the reports section is the possibility to generate graphs and statistics about what is happening in the domain, for example a graph about user activity or collaboration in the edition and creation of documents in Google Docs (see Figure 2). There are two links in every chart so it's possible to download the information in a table format or in the Comma-separated values (CSV) format. This is helpful because administrators can monitor resource usage and make decisions on service setup. The collaboration reports are especially useful in an educational institution, as they provide some information on collaborative behavior of people involved in the domain.

NCC/UFSM

The Federal University of Santa Maria (*Universidade Federal de Santa Maria - UFSM*) is a public Brazilian university located in the

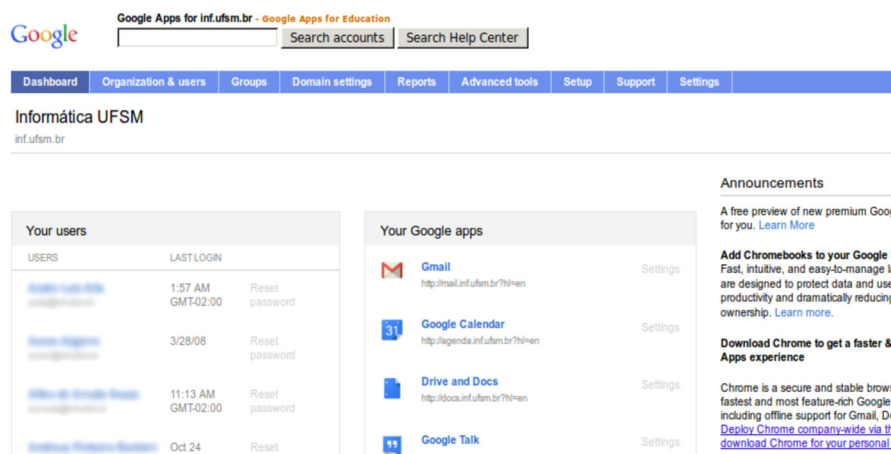


Figure 1. Google Apps control panel (Dashboard).

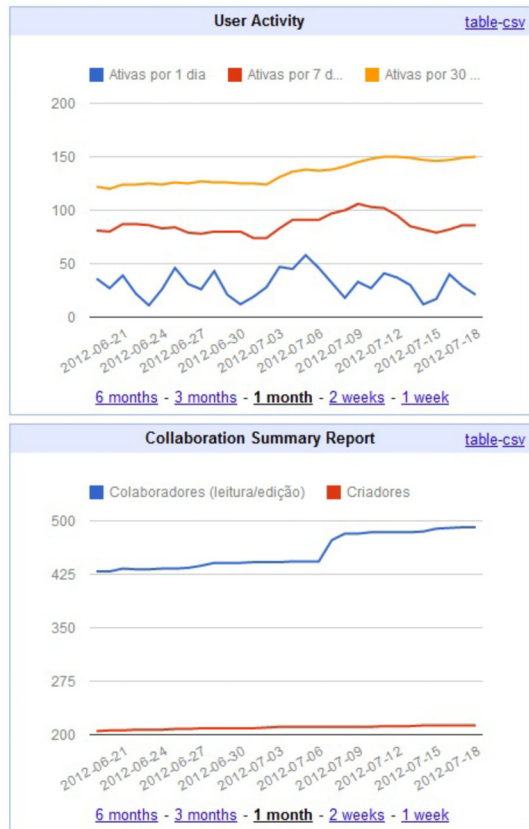


Figure 2. Example graphs and statistics.

city of Santa Maria. It is one of the largest universities of the state of Rio Grande do Sul, the southernmost state in Brazil. Currently, UFSM has more than 25.000 students and around 1.700 faculty members. It offers wide spectrum (more than 300) undergraduate and postgraduate courses.

NCC is one of the computer science laboratories attached to the Technology Center of UFSM. It was originated from the needs created by students and professors of the first undergraduate course on Information Technology at UFSM, which started in 1990. With this course, a demand for laboratories prepared with essential tools for teaching, research and extension in the field of computing started. Such laboratories comprise the NCC and are managed as a single organizational unit.

With the growth of the area in the institution, today the NCC serves undergraduate courses in Computer Science and Information Systems, and the first class started in 2009. The operation of NCC is guaranteed by faculty and student members. There is a coordinator (which is also a lecturer), a team of students with scholarships, and collaborators (faculty

or students). This allows students from both courses to participate in the NCC system administration, offering a learning opportunity and a real-world experience with networking technologies.

NCC's services and tasks have expanded over time and now include user management and support, management of network services (DNS, firewall, etc.) and virtualized servers, as well as managing all hardware and software resources. The volume of users grows regularly in the NCC, because every year a group of freshmen of Computer Science and Information Systems are registered in the domain and because the accounts of professors and students are kept even after they leave UFSM.

For some years, the NCC held a webmail service to its users, using the IMP tool (The Horde Project, 2012). IMP stands for Internet Messaging Program, which is a tool that provides webmail access to IMAP and POP3 accounts. This service demanded managing multiple servers (mail server, anti-spam, manager database, web server, etc.) and was a source of many problems, because it needed to be kept up to date and highly available. With the popularization of free email services, like Google Mail, many users started abandoning or redirecting their institutional email.

Given the popularization of webmail services in the cloud, many companies started to emerge and to offer these services, but at the time these types of companies weren't so common and many users were already using the Google Mail. In this scenario, Google Apps for Education emerged as an alternative to ensure webmail service availability to NCC users, while bringing other services aligned with the demands of NCC. Thus, in 2007, in a pioneering initiative within UFSM, the NCC began using this platform, after a migration process presented in the following section.

Migration process and current state

The first step to use Google Apps for Education is subscribing to the service, providing a domain name (in this case, inf.ufsm.br) and an email in this domain. In this step, Google checks the validity of the request to ensure that it is really a legitimate educational institution and that the applicant has permission on the domain. This step, in the NCC case, was completed relatively fast (about 24 hours).

To activate the service, and proving the permission on the domain, it was necessary to

put an entry in our DNS server, because some names needed to be redirected to Google's servers. In total, this initial process took a few days, after what we have been able to go to the next step: migration of user accounts.

At the time of the migration, to create a batch of users on Google Apps for Education, we needed to provide a spreadsheet with data for each user (name, surname, user name and password). At this point, each user gained an account in the cloud, but kept his/her local email account. The account credentials in the cloud were sent to users. Before those local services were disabled, we wanted to migrate data from emails (messages) that were stored locally. For this, Google offers a number of email migration options for both server and client sides (Google Inc., 2012e). We used the option that transfers the contents of the message boxes stored in IMAP server to the Google platform. This option could be used by individual users or in batch by the domain administrator. Initially, this option was placed at the disposal of each user, so that each one had control over migration options. For those who did not perform the migration on time, the automatic migration of the messages took place. From this moment, all the local email accounts were deactivated and only the accounts in the cloud remained. The data migration

step (messages on the email server) was the step that took longer, about a week to migrate approximately 30 GB.

After the migration process, we started using the interface of Google Apps for Education to manage users, groups, and other domain settings in the cloud. Over time, we have attained a limit number of users and it was necessary to request more accounts, which was answered promptly, without any cost. Currently, we have 900 registered users on Google Apps for Education in the inf.ufsm.br domain. Each year, since 2009, we created at least 80 new user accounts for freshmen of the Computer Science course (the first semester) and the Information Systems course (second semester), along with discussion groups for each new class.

The management team is configured via the control panel of Google Apps for Education, which lets one define roles (Super Admin, Admin Groups, User Management Admin, etc.) and privileges associated with each role (Create, Read, Rename, Move, Delete) (see Figure 3). One can also create new roles with custom permissions. This range of options fosters teamwork and aligns with the NCC organizational profile, where some management tasks are performed by students.

As creating users and groups are recurring tasks, the web-based control panel has

The screenshot shows the Google Apps for Education interface for the domain 'inf.ufsm.br'. The 'Domain settings' page is active, with the 'Admin roles' tab selected. On the left, a sidebar lists 'System Roles' and 'User Created Roles'. The 'Super Admin' role is highlighted. The main content area shows the 'Admins' and 'Privileges' for the 'Super Admin' role. A checkbox indicates 'This user has full Administrative rights within Organization and Users'. Below this, two tables show the privileges for 'Organizations General' and 'User General'. Both tables have columns for 'Create', 'Read', 'Rename', 'Move', and 'Delete', all of which are checked.

Organizations General		Create	Read	Rename	Move	Delete
		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

User General		Create	Read	Rename	Move	Delete
		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 3. Domain settings: roles and privileges.

proved to be a bit inefficient, as it requires a lot of human intervention in a point-and-click interface. Even if some tasks may be performed in batch through the control panel (e.g. bulk upload users), this is not sufficient to help us automate some of our main tasks. As mentioned before, Google Apps provides APIs and other resources for systems administrators. Thus, we have been studying alternatives to automate the management of groups and accounts. In particular, we have been exploring some opportunities provided by two Google Apps resources: Google Provisioning API and Google Apps Directory Sync.

Opportunities

In this section, we describe our experience with the aforementioned resources provided by Google Apps as possible helpers for two recurrent tasks: managing user groups and synchronizing accounts between Google Apps and our local LDAP service. For each task, we first describe our motivation and goals and then present some information on the Google Apps resource that we are exploring to best meet our requirements.

Managing groups with Google provisioning API

Google Groups is one of the services included in the Google Apps platform. A group comprises a list of email addresses (inside or outside the inf.ufsm.br domain). Groups can be used for communication, collaboration and resource sharing in the Google Apps platform.

In our organization, there are currently around 80 groups. A few of them are permanent groups, in the sense they refer to organizational groups that do not change their names or composition very often. Examples

are the faculty group, the system administrators group, among others. Most of the groups, however, have a shorter existence, as for example the groups for undergraduate courses, each comprising of a professor and its students for a semester.

Currently, all users are allowed to create groups using the Google Apps web interface. Group management options are also available to group owners, who can manually edit group memberships and related options. However, for certain short-term groups, namely the groups for undergraduate courses, we found this manual procedure inefficient. Indeed, many professors end up not creating groups and, when they create, they usually do not remove the group after the end of the course/semester. In such cases, we believe that an automated management could be more efficient.

Using Google Apps APIs, it is possible to create programs that automate procedures originally performed by hand. In this context, an important resource is Google Provisioning API (Google Inc., 2012h), which allows to programmatically manage users and groups in a Google Apps domain. Implementations of this API are available in various languages like Java, Python, NET and PHP. There is also a comprehensive documentation on this API, including information on service protocols, supporting languages and examples.

To start working with Google Provisioning API, it is necessary first to enable it in the Google Apps control panel for the inf.ufsm.br domain. By default, Google Apps enables only a few services and does not allow API access without permission from the domain manager.

Using Google Provisioning API, the communication between a client program and the domain server is very simple. It requires a two-way authentication, which means that the client sends a request and the server responds

```
String adminEmail="admin_email@inf.ufsm.br";
String PASSWORD = "admin_password";

try {
    /* client authentication */
    client = new AppsForYourDomainClient(adminEmail, PASSWORD, "inf.ufsm.br");

    /* getting the groups service */
    groupService = client.getGroupService();
}
```

Figure 4. Authentication with pure text.

indicating whether the client is authenticated and authorized or not. After authenticated in the domain, the client can invoke all the methods it is authorized to.

For example, to create a group, the client program has to authenticate (see Figure 4), then it uses the method for creating a group (see Figure 5a) and add members to the recently created group (see Figure 5b). Of course, there are more methods available, as for example the method for removing users from a group (see Figure 5c).

Using Google Provisioning API, we developed a set of programs that manage groups given one or more of its member's email addresses. Our main use case is the automatic creation of groups for undergraduate courses each semester, based on a list of students provided by the university information system. The programs can support similar use cases; the only requisites being the group name and the email address(es).

We decided to create four programs, written in Java, to perform the following tasks: (i) add one member in an existing group, (ii) remove one member from a group, (iii) create a group based on a list of users and (iv) update an existing group based on a list of users. All the programs run as command-line tools, so is possible to create a script that automatically calls these programs in the beginning of the semester, for example. After the first groups are created, another script may run the update program, because sometimes students give up their courses, or register in different courses and this causes an inconsistency with the recently created groups.

While developing these solutions, we faced a few implementation issues that are worth mentioning. The first issue concerns the authentication method. We started using a simple method that only required a user account and password. The only problem with that method was that the password is a pure text in the source code (see Figure 4), so if someone

gets the source code of any of the programs, he or she can access this email account. This problem can be minimized by creating a fictitious account and assigning a role to the account with a series of permissions. We also investigated an alternative method using the OAuth authentication protocol (OAuth Community Site, 2012) that provides access to only a specific resource and for just a certain amount of time. Using this authentication method, the user name and the password would not be exposed. However, the process to implement this type of authentication proved to be a little bit confusing and, in our case, we decided that the simplest solution was the best for a first version of the programs recently created.

The second issue we faced concerns gathering the input data for our programs. In our main use case, the input data is a list of user accounts identifying students that are enrolled in a given course in a given semester. All this data is held by an information system maintained by the central academic administration of UFSM, so we needed to retrieve that data and perform some preprocessing before providing it as input to our programs. The main issue at this point is that the account names are only known in NCC, so after we retrieve the student names, we have to match that information with our LDAP database to get the account names. This proved to be a problem, because sometimes the user names in our LDAP database are not the same as the user names registered in the inf.ufsm.br domain in Google Apps. This problem led us to study a bit further the Google Apps service for synchronizing our cloud-based domain with our local LDAP database.

Synchronizing accounts with Google Apps Directory Sync

Regarding the creation of user accounts in NCC, we currently perform, each semester, a two-step procedure for each new group of

```
a) groupService.createGroup(groupId, groupName,
                             groupDescription, permissions);

b) groupService.addMemberToGroup(groupName, user);

c) groupService.deleteMemberFromGroup(groupName, user);
```

Figure 5. Examples of methods of the Provisioning API in Java.

freshmen: (a) create user accounts on our local LDAP database, which is queried by local services (access to laboratory machines, remote access through SSH, etc.), and (b) create these user accounts in our cloud-based domain in Google Apps platform. The second step is usually performed using the “bulk upload users” option provided by Google Apps. This two-step procedure, however, requires human intervention and is not performed as an atomic transaction. This approach led to some inconsistencies between our local and cloud-based services.

Recently, Google introduced a utility that could help us in this scenario: Google Apps Directory Sync (GADS). The goal of this utility is to automatically update a Google Apps domain to match a local LDAP database. GADS is distributed as a standalone application which runs on the local LDAP server and communicates with Google Apps servers. This utility is multi-platform, with versions for Windows, Linux and Solaris, for 32 or 64 bits. It supports most generic Open LDAP servers, besides Microsoft Active Directory and Lotus Domino.

GADS is actually composed of two related tools: a GUI-based wizard which helps in configuring a synchronization (see Figure 6) and a command-line tool that reads the configuration and actually performs a synchronization. The command-line tool can be scheduled to run from time to time, keeping equivalent entries in both cloud and local servers, without the need to duplicate work.

The command-line tool performs one-way synchronization of accounts, groups and other data items currently maintained in an LDAP server. The LDAP schema in the local server is not updated or altered by GADS. The utility connects to the local LDAP server and generates a list of users, groups, and shared contacts

from the LDAP directory. Using the GUI-based wizard, one can specify a set of rules configuring how this list is generated. The utility also connects to Google Apps and generates a list of users, groups and contacts that have been created on the platform. Again, the generation of the list can be customized through rules. A comparison is made between the local and remote data so that the Google Apps data items become identical to the local LDAP directory. After synchronization completes, a report is sent with the results to pre-configured email addresses.

GADS offer several additional features, such as: synchronization tests to avoid failures, scheduling synchronization of directories, enabling automation of updating the database, limitation of changes based on rules and permissions, along with automatic notifications to all users. These are just some of the automated services that streamline the administration of the database.

In the NCC case, we are mainly interested on synchronizing accounts, i.e., user names and passwords. As mentioned before, our LDAP directory is primarily used to authenticate users in our local services. It does not maintain detailed profile data about the users. Permanent group memberships are stored in our local base, but short-term groups are only registered in Google Apps. Concerning passwords, there are some limitations on the hash encoding formats supported by GADS.

Currently, the use of GADS is still in analysis in NCC, because some potential synchronization issue could undermine the existing registration of users on the platform. One of our main concerns is about passwords, because in our current environment, NCC users have two passwords: one for local services and one for

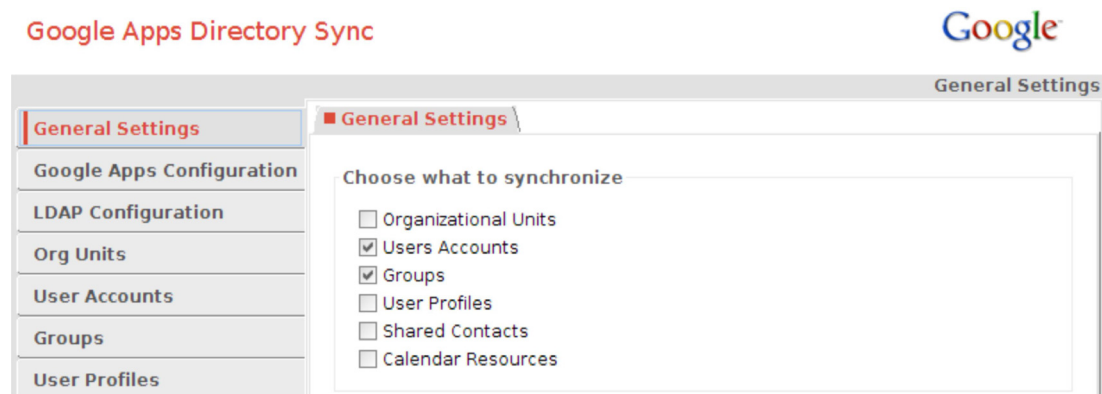


Figure 6. GADS synchronization helping wizard.

Google Apps. In this scenario, we will need to set up some rules and carefully manage the synchronization process, so NCC users do not suffer with password changes. Although we are not yet running GADS, we can definitely say that there are many positive aspects about this utility.

Concluding remarks

This experience report supported the idea that cloud platforms can effectively help educational institutions in their support activities, without much effort.

By migrating the inf.ufsm.br domain to the Google Apps for Education cloud platform, NCC achieved the goal of having various services being provided to its users with higher availability, compared to when these services were provided by local servers. In our experience, such migration has benefited both students and faculty involved.

Google Apps for Education also offers services and APIs for system administrators. These are valuable resources for a computer science laboratory in an educational institution, as it allows for students and faculty to stay involved in local development activities for customizing and improving the environment. We believe that this experience might contribute to other institutions similar to NCC in assessing the possibility of migrating services to a cloud platform.

References

- BARLOW, K.; LANE, J. 2007. Like technology from an advanced alien culture: Google apps for education at asu. In: ANNUAL ACM SIGUCCS FALL CONFERENCE, 35, New York, 2007. *Proceedings...* ACM, p. 8-10. Available at: <http://doi.acm.org/10.1145/1294046.1294049>. Accessed on: 10/22/2012.
- GOOGLE INC. 2012a. Pricing: Google Apps for Business. Available at: <http://www.google.com/enterprise/apps/business/pricing.html>. Accessed on: 10/22/2012.
- GOOGLE INC. 2012b. Google Apps for Education. Available at: <http://www.google.com/apps/edu>. Accessed on: 10/22/2012.
- GOOGLE INC. 2012c. Google Apps. Available at: <http://www.google.com/apps/intl/en/index.html>. Accessed on: 10/22/2012.
- GOOGLE INC. 2012d. Success stories: Google Apps for Education. Available at: <http://www.google.com/apps/intl/en/edu/customers.html>. Accessed on: 10/22/2012.
- GOOGLE INC. 2012e. E-mail migration options: Google Apps help. Available at: <http://support.google.com/a/bin/answer.py?hl=en&answer=57920>. Accessed on: 10/22/2012.
- GOOGLE INC. 2012f. Google Apps help. Available at: <http://www.google.com/apps/intl/en/edu/customers.html>. Accessed on: 10/22/2012.
- GOOGLE INC. 2012g. Google Apps Directory Sync. Available at: <http://support.google.com/a/bin/answer.py?hl=en&answer=106368>. Accessed on: 10/22/2012.
- GOOGLE INC. 2012h. Google Apps Provisioning API. Available at: <https://developers.google.com/google-apps/provisioning/>. Accessed on: 10/22/2012.
- HERRICK, D.R. 2009. Google this!: using Google apps for collaboration and productivity. In: ANNUAL ACM SIGUCCS FALL CONFERENCE, 37, New York, 2009. *Proceedings...* ACM, p. 55-64. Available at: <http://doi.acm.org/10.1145/1629501.1629513>. Accessed on: 10/22/2012.
- OAuth COMMUNITY SITE. 2012. What is it For? Available at: <http://oauth.net/about/>. Accessed on: 10/22/2012.
- SULTAN, N. 2010. Cloud computing for education: A new dawn? In: International Journal of Information Management, vol. 30, no. 2, pp. 109 – 116, 2010. Available at: <http://www.sciencedirect.com/science/article/pii/S0268401209001170>. Accessed on: 10/22/2012.
- THE HORDE PROJECT. 2012. IMP. Available at: <http://www.horde.org/apps/imp/>. Accessed on: 10/22/2012.

Submitted on October 10, 2012

Accepted on December 11, 2012